

Język PL/SQL. Rozdział 6.

Wyzwalacze bazy danych – zadania

1. Napisz wyzwalacz, który będzie automatycznie przyznawał kolejne identyfikatory nowym zespołom. Wartości dla identyfikatorów powinny być generowane przez sekwencję. Przetestuj działanie wyzwalacza z poniższymi poleceniami.

```
SQL> INSERT INTO ZESPOLY (NAZWA) VALUES ('KRYPTOGRAFIA');
```

1 wiersz został utworzony.

```
SQL> INSERT INTO ZESPOLY (NAZWA) SELECT substr('NOWE '||NAZWA,1,20) FROM ZESPOLY
WHERE ID_ZESP in (10,20);
```

2 wiersze zostały utworzone.

2. Dodaj do relacji ZESPOLY atrybut LICZBA_PRACOWNIKOW. Napisz zlecenie SQL które zainicjuje początkowe wartości atrybutu. Napisz wyzwalacz **wierszowy**, który będzie pielęgnował wartość tego atrybutu. Przetestuj działanie wyzwalacza.

```
SQL> ALTER TABLE ZESPOLY ADD (LICZBA_PRACOWNIKOW NUMBER);
```

```
SQL> UPDATE ZESPOLY Z
SET LICZBA_PRACOWNIKOW =
(SELECT COUNT(*) FROM PRACOWNICY WHERE ID_ZESP = Z.ID_ZESP);
```

```
SQL> SELECT * FROM zespoly;
```

ID_ZESP	NAZWA	ADRES	LICZBA_PRACOWNIKOW
10	ADMINISTRACJA	PIOTROWO 3A	2
20	SYSTEMY ROZPROSZONE	PIOTROWO 3A	7
30	SYSTEMY EKSPERCKIE	STRZELECKA 14	4
40	ALGORYTMY	WLODKOWICA 16	1
50	BADANIA OPERACYJNE	MIELZYNSKIEGO 30	0

```
SQL> INSERT INTO pracownicy (ID_PRAC, NAZWISKO, ID_ZESP, ID_SZEFA)
VALUES (300, 'NOWY PRACOWNIK', 40, 120);
```

1 wiersz został utworzony.

```
SQL> SELECT * FROM zespoly;
```

ID_ZESP	NAZWA	ADRES	LICZBA_PRACOWNIKOW
10	ADMINISTRACJA	PIOTROWO 3A	2
20	SYSTEMY ROZPROSZONE	PIOTROWO 3A	7
30	SYSTEMY EKSPERCKIE	STRZELECKA 14	4
40	ALGORYTMY	WLODKOWICA 16	2

```
SQL> UPDATE PRACOWNICY SET ID_ZESP = 10 WHERE ID_ZESP = 30;
```

4 wiersze zostały zmodyfikowane.

```
SELECT * FROM zespol;
```

ID_ZESP	NAZWA	ADRES	LICZBA_PRACOWNIKOW
10	ADMINISTRACJA	PIOTROWO 3A	6
20	SYSTEMY ROZPROSZONE	PIOTROWO 3A	7
30	SYSTEMY EKSPERCKIE	STRZELECKA 14	0
40	ALGORYTMY	WLODKOWICA 16	0

3. Zdefiniuj relację HISTORIA o schemacie (ID_PRAC, PLACA_POD, ETAT, ZESPOL, MODYFIKACJA). Napisz wyzwalacz, który po każdej modyfikacji wartości płacy podstawowej, etatu lub zespołu w relacji PRACOWNICY będzie wpisywał wartości historyczne do relacji HISTORIA (wartości sprzed modyfikacji).

```
SQL> CREATE TABLE HISTORIA (  
    ID_PRAC NUMBER,  
    PLACA_POD NUMBER,  
    ETAT VARCHAR2(20),  
    ZESPOL VARCHAR2(20),  
    MODYFIKACJA DATE);
```

```
SQL> UPDATE PRACOWNICY SET PLACA_POD = 800 WHERE NAZWISKO = 'KROLIKOWSKI';
```

```
SQL> DELETE FROM PRACOWNICY WHERE NAZWISKO = 'HAPKE';
```

```
SQL> SELECT * FROM HISTORIA;
```

ID_PRAC	PLACA_POD	ETAT	ZESPOL	MODYFIKACJ
150	658,41	ADIUNKT	SYSTEMY ROZPROSZONE	13-12-2001
230	480	ASYSTENT	SYSTEMY EKSPERCKIE	13-12-2001

4. Zdefiniuj perspektywę SZEFOWIE(SZEF, PRACOWNICY) zawierającą nazwisko szefa i liczbę jego podwładnych. Napisz procedurę wyzwalaną która umożliwi, za pomocą powyższej perspektywy, usuwanie szefów wraz z kaskadowym usunięciem wszystkich podwładnych danego szefa. Jeśli podwładny usuwanego szefa sam jest szefem innych pracowników, przerwij działanie wyzwalacza błędem o numerze ORA-20001 i komunikacie „Jeden z podwładnych usuwanego pracownika jest szefem innych pracowników. Usuwanie anulowane!”.

```
SQL> SELECT * FROM szefowie;
```

SZEF	PRACOWNICY
BLAZEWICZ	1
BRZEZINSKI	5
MORZY	2
SLOWINSKI	2
WEGLARZ	3

```
SQL> SELECT * FROM pracownicy WHERE id_prac = 140 OR id_szefa = 140;
```

ID_PRAC	NAZWISKO	ETAT	ID_SZEFA	ZATRUDNI	PLACA_POD	PLACA_DOD	ID_ZESP
140	MORZY	PROFESOR	130	75/09/15	830	105	20
190	MATYSIAK	ASYSTENT	140	93/09/01	371		20
200	ZAKRZEWICZ	STAZYSTA	140	94/07/15	208		30

```
SQL> DELETE FROM szefowie WHERE szef='MORZY';
```

```
SQL> SELECT * FROM pracownicy WHERE id_prac = 140 OR id_szefa = 140;
```

nie wybrano żadnych wierszy

Przywróć usunięte rekordy wycofując poleceniem ROLLBACK transakcję, w której nastąpiło usunięcie pracownika MORZY.

5. W relacji PRACOWNICY usuń ograniczenie referencyjne FK_ID_SZEFA (klucz obcy między pracownikiem a jego szefem), następnie utwórz je ponownie z cechą usuwania kaskadowego.

```
SQL> ALTER TABLE PRACOWNICY DROP ...;
```

```
SQL> ALTER TABLE PRACOWNICY ADD ...;
```

Zdefiniuj teraz wyzwalacz wierszowy o nazwie USUN_PRAC. Wyzwalacz ma uruchamiać się po wykonaniu operacji DELETE na relacji PRACOWNICY. Jedynym zadaniem wyzwalacza będzie wypisanie na ekranie, za pomocą procedury DBMS_OUTPUT.PUT_LINE, nazwiska usuwanego pracownika. Przetestuj działanie wyzwalacza usuwając z relacji PRACOWNICY rekord opisujący pracownika o nazwisko MORZY. Nie zapomnij przed wykonaniem polecenia DELETE ustawić zmiennej SERVEROUTPUT na wartość ON. Po zakończeniu zadania wycofaj transakcję przy pomocy polecenia ROLLBACK;

Wykonaj ponownie zadanie 5. Tym razem wyzwalacz USUN_PRAC ma się uruchamiać przed wykonaniem operacji DELETE na relacji PRACOWNICY. Porównaj otrzymane teraz wyniki z wynikami z pierwszej części zadania.